IDS 561 - Final Project Report

Netflix: Recommendation System

Team 12 = Raymond Sutanto + Cheng Lin Tsai + Yi Jen Chen + Stanley Lin

☞ **Problem Setting**

As we finish seeing a movie or a TV show on Netflix, it will suggest additional movies and TV series that you might enjoy watching. This is made possible via Netflix's Recommendation System, which comprises various algorithms and machine learning. Netflix uses machine learning and algorithms to help consumers find shows they would not have chosen on their own and to challenge their prejudices. Instead of relying on broad categories to make predictions, it looks for small links within the material. As a result, we have divided our issue into multiple challenges. First, we'll look at how Netflix's Recommendation System works. Second, we'd like to learn about the underlying criteria that Netflix uses to choose movie and TV program suggestions for viewers. Lastly, we'd want to go at the underlying algorithms and machine learning in this system, as well as some of the efforts Netflix has made to enhance it.

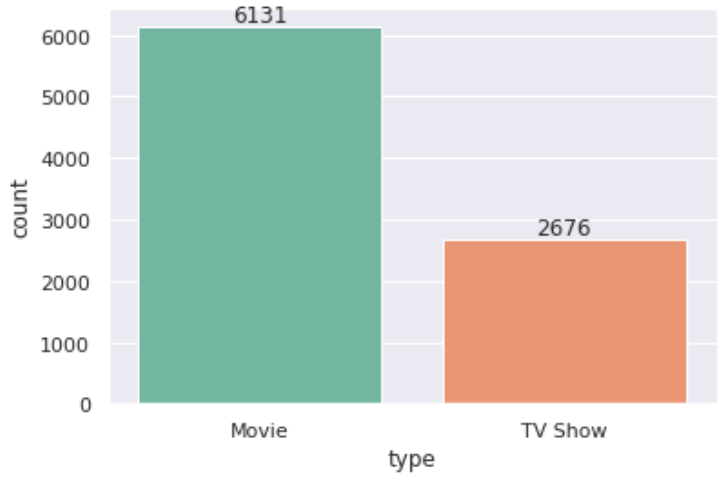
☞ **Data Description**

The dataset that we are working with was obtained from Kaggle, a popular platform for data science and machine learning enthusiasts. It contains information on various movies and TV programs, and each entry in the dataset is associated with a unique ID. Along with the ID, there are other important attributes associated with each entry, such as the type of content (movie or TV program), the title of the content, the director who was involved in creating it, the cast of actors, the country in which it was produced, the date it was uploaded to Netflix, the year it was released, the rating given by viewers, and the runtime of the content.

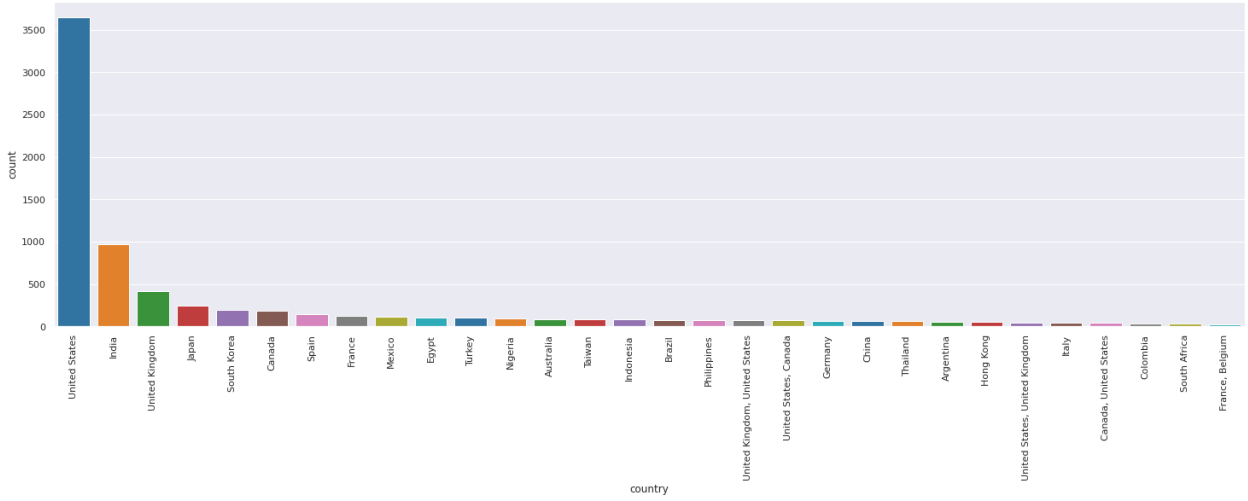
The dataset being described consists of approximately 8800 entries with 12 different attributes. These attributes include show_id, type, title, director, cast, country, date_added, release_year, rating, duration, listed_in, and description. The show_id is a unique identifier assigned to each entry, while the type attribute indicates whether the entry is a movie or a TV show. The title attribute is the name of the movie or TV show, and the director and cast attributes provide information on the people involved in creating the movie or TV show. The country attribute indicates the country where the movie or TV show was produced, and the date_added attribute indicates the date when the movie or TV show was added to the dataset. The release_year attribute provides the year when the movie or TV show was originally released, and

the rating attribute indicates the appropriate age rating for the movie or TV show. The duration attribute specifies the length of the movie or TV show, and the listed_in attribute provides information on the genre or category of the movie or TV show. Finally, the description attribute offers a brief summary of the movie or TV show. This dataset can be used for various purposes, such as analyzing trends in the entertainment industry or building recommendation systems.

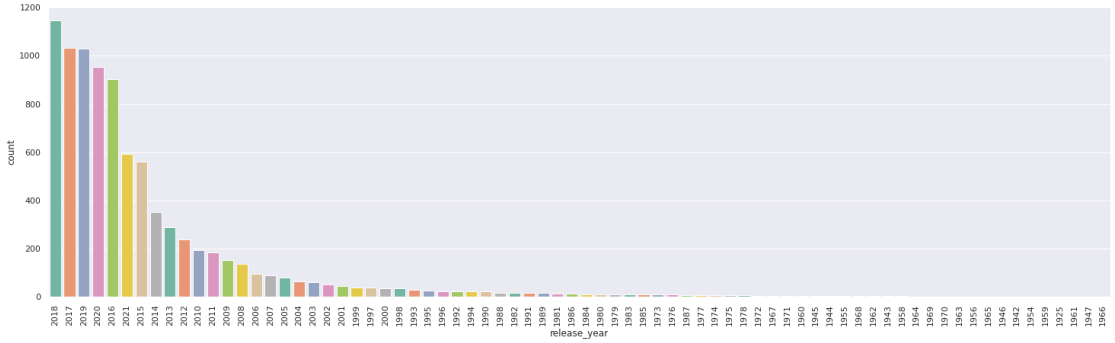
Below is the chart showing the amount of people who watch movies the most or TV shows.



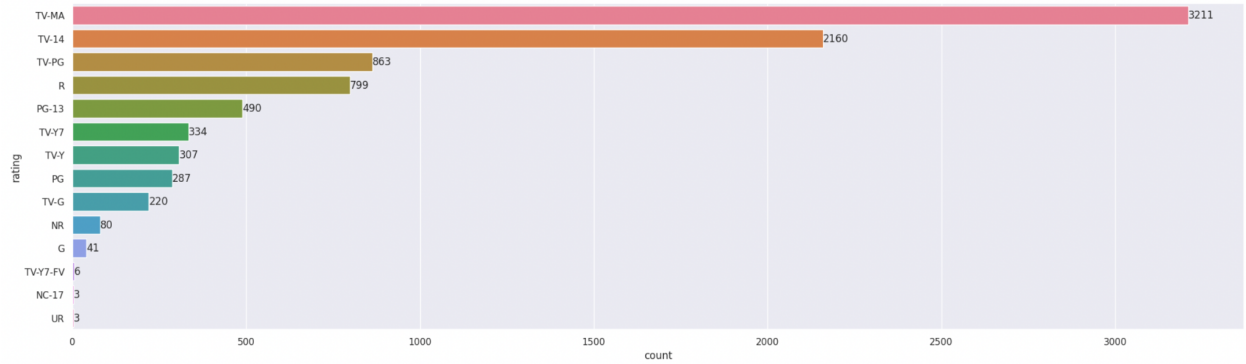
Below is the bar plot showing the amount of content that originated from each country.



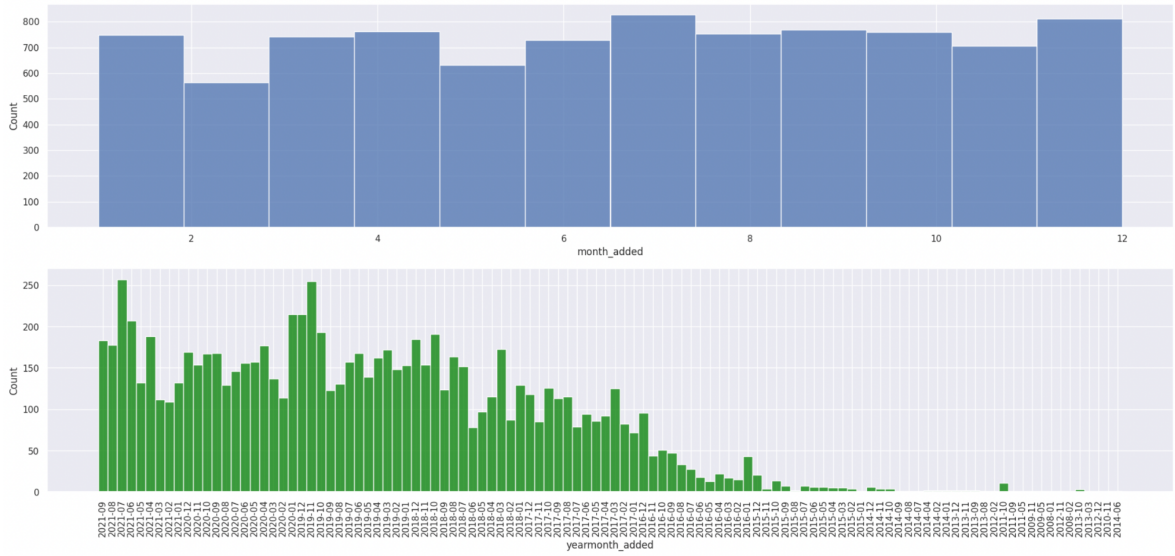
Below is the bar plot showing the amount of content that is made in the year.



Below is the bar plot showing all kinds of TV Shows or Movies types in these years.



Below is the bar plot showing the amount of TV Shows and Movies added in Netflix monthly and yearly.



☞ **Techniques**

Content-based filtering is a powerful technique used in recommender systems to provide users with personalized recommendations. This approach is based on analyzing the attributes and features of the items that the user has previously liked or interacted with. These attributes can include factors such as the title, cast, director, listed_in, and plot of a movie, among others. By using this information, we can identify patterns and similarities between different items and provide recommendations that are tailored to the user's preferences.

To implement this approach, we use a technique called Term Frequency-Inverse Document Frequency (TF-IDF). This technique computes a score for each word in a document based on how frequently it occurs in that document, while also taking into account how common the word is across all documents in the dataset. By down-weighting words that occur frequently in plot overviews, we can ensure that the final similarity score is based on the most relevant and distinctive features of each movie.

```
#Importing TF-IDF
from sklearn.metrics.pairwise import linear_kernel
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix_desc=tfidf.fit_transform(df['description'])
tfidf_matrix_genre=tfidf.fit_transform(df['listed_in'])
#Deploy cosine similarity
cosine_sim_desc = linear_kernel(tfidf_matrix_desc,tfidf_matrix_desc)
cosine_sim_genre = linear_kernel(tfidf_matrix_genre,tfidf_matrix_genre)
cosine_sim_genre
```

Our dataset contains approximately 16,000 words describing around 6,000 movies, providing a rich and detailed set of attributes for our content-based filtering algorithm to analyze. By leveraging this wealth of information, we can provide users with personalized recommendations that are based on the specific attributes and features that they have shown an interest in. Ultimately, this approach can help users discover new and exciting movies that they may not have otherwise known about, while also helping to ensure that their viewing experience is tailored to their individual tastes and preferences.

When it comes to finding similarity between vectors, cosine similarity is a widely used technique here. Its independence from the magnitude of the vectors being compared makes it particularly useful in text mining and information retrieval applications. The magnitude of a vector is simply the length of the vector in space, and in text mining, this can vary significantly depending on the length of the documents being compared. By ignoring the magnitude of the vectors and only considering their direction, cosine similarity ensures that the similarity score remains consistent, regardless of the magnitude of the vectors being compared. Below graph shows how we use the cosine similarity to measure the similarity scores by implementing the description of recommendation systems.

```
#Define get recommendation for description
def get_recommendations_desc(title):
    idx = lists[title]
    sim_scores_desc = list(enumerate(cosine_sim_desc[idx]))
    sim_scores_desc = sorted(sim_scores_desc, key=lambda x: x[1], reverse=True)
    sim_scores_desc = sim_scores_desc[1:11]
    movie_lists = [i[0] for i in sim_scores_desc]
    return df[['title','description','listed_in']].iloc[movie_lists]
```

```
get_recommendations_desc("Stranger Things")
```

	title	description	listed_in
4733	Rowdy Rathore	A con man uncovers a deadly secret and must sa...	Action & Adventure, Comedies, International Mo...
1240	Safe Haven	When a mysterious woman arrives in a small Nor...	Dramas, Romantic Movies
1487	Sakho & Mangane	A by-the-book police captain and a brash young...	Crime TV Shows, International TV Shows, TV Dramas
8198	The Autopsy of Jane Doe	A father-son team of small-town coroners perfo...	Horror Movies, Independent Movies, Thrillers
2419	Big Stone Gap	A single middle-aged woman who has lived her w...	Comedies, Romantic Movies
6518	Come and Find Me	When his photographer girlfriend vanishes, an ...	Dramas, Thrillers
6760	FirstBorn	A young couple fights supernatural foes in an ...	Horror Movies, International Movies
8026	Sinister Circle	A psychologist and her mute son confront evil ...	Horror Movies, International Movies
4201	Hardy Bucks	A circle of young men entertain vague ambition...	TV Comedies
1270	Sin senos sÃ hay paraÃso	Born into a small town controlled by the mafia...	International TV Shows, Spanish-Language TV Sh...

In addition to this key advantage, cosine similarity is also relatively easy and fast to compute, making it an ideal choice for large-scale recommendation systems. In these systems, where efficiency is paramount for handling large volumes of data, the ability to quickly calculate similarity scores between thousands or even millions of items is essential. By leveraging the simplicity and speed of cosine similarity, we can efficiently generate personalized recommendations for users, even when working with massive datasets.

We deploy the Natural Language Toolkit (NLTK) library for Python that downloads a collection of popular datasets and resources used for natural language processing tasks. These datasets and resources include corpora, which are large collections of text that are often used for training and testing machine learning models, as well as various tools and libraries for tasks such as stemming, tokenization, and part-of-speech tagging. Executing this command enables us to promptly acquire numerous frequently utilized resources with just a single command instead of manually downloading each one. This feature can effectively reduce the time and effort required when initializing a new NLP project or working with unfamiliar datasets.

```
def preprocess(text):
    text = text.lower()
    text = re.sub('[^A-z]', ' ', text)
    stop_words = set(stopwords.words('english'))
    word_tokens = word_tokenize(text)
    lemmatizer = nltk.stem.WordNetLemmatizer()

    filtered_sentence = []
    for w in word_tokens:
        if w not in stop_words:
            filtered_sentence.append(lemmatizer.lemmatize(w))
    filtered = ' '.join([x for x in filtered_sentence])
    return filtered.lower().strip()
```

```
df_final['cleaned_text'] = df_final['text'].apply(lambda x : preprocess(x))
```

The 'preprocess' function (see the graph above) plays a crucial role in natural language processing (NLP) by performing a series of essential text preprocessing operations. These operations include transforming the text to lowercase, eliminating non-alphabetic characters, tokenizing the text into individual words or phrases, removing stopwords or commonly used words such as 'the', 'a', and 'an', and lemmatizing words to their base form. By applying these preprocessing techniques, the 'preprocess' function ensures that the text is in a consistent format that can be effectively used in NLP applications such as text classification and sentiment analysis. Lowercasing the text, for example, ensures that the same word in different cases is treated as the same entity, thus improving accuracy.

Removing non-alphabetic characters and stopwords helps in reducing the overall noise in the text data, allowing the model to focus on more significant features. Tokenization breaks down the text into individual components, which makes it easier for the model to understand the context of the text. Finally, lemmatizing the text ensures that words are represented in their base form, simplifying the text and reducing the number of features that the model has to deal with, improving efficiency.

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
text_features = vectorizer.fit_transform(df_final['cleaned_text'])
```

```
from sklearn.metrics.pairwise import cosine_similarity
similarity_matrix = cosine_similarity(text_features)
```

```
#The final output to get the recommendation by film's title
def get_recommendations(title):
    idx = lists[title]
    sim_scores = list(enumerate(similarity_matrix[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:11]
    movie_lists = [i[0] for i in sim_scores]
    return df_final[['title','listed_in','description']].iloc[movie_lists]
```

After we cleanse the dataset as a final source to get the recommendation system, we apply another advantage of cosine similarity with its flexibility in handling different types of data. In recommendation systems, cosine similarity is an important tool for measuring the similarity between items. By representing items as vectors, cosine similarity can be used to measure the angle between them, which provides a measure of their similarity. This method is widely used in recommendation systems because it is highly effective at capturing the

relationships between items, especially when dealing with large and complex datasets. Moreover, cosine similarity is also highly scalable, which means that it can be used to analyze millions of data points in a matter of seconds.

The versatility of cosine similarity is also a key advantage in recommendation systems. For instance, it can be used to compare not only the content of movies and TV shows, but also other attributes such as genre, director, actors, and so on. In this way, it can provide highly personalized recommendations based on a user's preferences, without being restricted to a particular type of data. Overall, cosine similarity is a powerful and flexible tool that is well-suited to the task of recommendation systems, and is likely to remain an important part of data analytics for years to come.

☞ Results & Conclusion

```
get_recommendations("Stranger Things")
```

	title	listed_in	description
5200	Beyond Stranger Things	Stand-Up Comedy & Talk Shows, TV Mysteries, TV...	Secrets from the "Stranger Things 2" universe ...
1127	Prank Encounters	Reality TV, TV Comedies, TV Horror	Monstrous frights meet hilarious reveals on th...
2190	The Umbrella Academy	TV Action & Adventure, TV Mysteries, TV Sci-Fi...	Reunited by their father's death, estranged si...
6167	Anjaan: Special Crimes Unit	International TV Shows, TV Horror, TV Mysteries	The cases are supernatural; the police officer...
1335	The Sinner	Crime TV Shows, TV Dramas, TV Mysteries	When a young mother inexplicably stabs a stran...
964	Things Heard & Seen	Horror Movies, Thrillers	A young woman discovers that both her husband ...
3986	The OA	TV Dramas, TV Mysteries, TV Sci-Fi & Fantasy	Seven years after vanishing from her home, a y...
4809	Kiss Me First	British TV Shows, Crime TV Shows, Internationa...	A lonely young woman hooked on a virtual reali...
8604	Top 10 Secrets and Mysteries	British TV Shows, Docuseries, Science & Nature TV	This series investigates mysteries that persis...
241	Manifest	TV Dramas, TV Mysteries, TV Sci-Fi & Fantasy	When a plane mysteriously lands years after ta...

```
get_recommendations("Friends")
```

	title	listed_in	description
5090	Episodes	Classic & Cult TV, TV Comedies	Hoping to create an American version of their ...
1929	Man with a Plan	TV Comedies	When his wife Andi returns to work, contractor...
6549	Dad's Army	British TV Shows, Classic & Cult TV, TV Comedies	This beloved sitcom follows the unlikely heroe...
7301	Life Story	British TV Shows, Docuseries, Science & Nature TV	This documentary series follows wild creatures...
4096	Studio 54	Documentaries	This documentary follows the rapid rise and fa...
7397	Manhattan Romance	Comedies, Independent Movies, Romantic Movies	A filmmaker working on a documentary about lov...
2080	Thomas and Friends	British TV Shows, Classic & Cult TV, Kids' TV	This animated children's series follows the ad...
682	Why Are You Like This	International TV Shows, TV Comedies	Three best friends negotiate work, fun, identi...
3898	Lunatics	International TV Shows, TV Comedies	This mockumentary series follows the peculiar ...
4265	Single Ladies Senior	International TV Shows, Romantic TV Shows, TV ...	Four best friends and spirited career women na...

```
get_recommendations("A Classic Horror Story")
```

	title	listed_in	description
1898	The Binding	Horror Movies, International Movies, Thrillers	While visiting her fiancÃ©'s mother in souther...
3924	Francesco De Carlo: Cose di Questo Mondo	Stand-Up Comedy	Taking the stage in Milan, Francesco De Carlo ...
8208	The Bible's Buried Secrets	British TV Shows, Docuseries, Science & Nature TV	Host Francesca Stavrakopoulou travels across t...
7537	My Honor Was Loyalty	Dramas	Amid the chaos and horror of World War II, a c...
8188	The Amityville Horror	Horror Movies	This hair-raising remake of the 1979 horror hi...
1571	Rose Island	Comedies, Dramas, International Movies	An idealistic engineer builds his own island o...
2974	Luna Nera	International TV Shows, Romantic TV Shows, TV ...	In 17th-century Italy, a teenager learns about...
2108	Goedam	International TV Shows, TV Horror, TV Mysteries	When night falls on the city, shadows and spir...
5852	Hush	Horror Movies, Thrillers	A deaf writer who retreated into the woods to ...
3548	Feo pero sabroso	Comedies, International Movies	When an unattractive man gets engaged to a bea...

```
get_recommendations("The Umbrella Academy")
```

	title	listed_in	description
8316	The Future of Water	Docuseries, International TV Shows, Science & ...	A look at the deeply intertwined history of hu...
478	Biohackers	International TV Shows, TV Dramas, TV Sci-Fi &...	A medical student enters a top German universi...
2303	Warrior Nun	TV Action & Adventure, TV Mysteries, TV Sci-Fi...	After waking up in a morgue, an orphaned teen ...
5939	The 4400	TV Dramas, TV Mysteries, TV Sci-Fi & Fantasy	4400 people who vanished over the course of fi...
3246	The Dragon Prince	Kids' TV, TV Action & Adventure, TV Sci-Fi & F...	An extraordinary discovery inspires two human ...
7381	Maharakshak Devi	International TV Shows, TV Action & Adventure,...	After years spent in isolation, a young girl w...
7382	Maharakshak: Aryan	International TV Shows, TV Action & Adventure,...	Shy nerd Aryan learns soon after his 18th birt...
993	Shadow and Bone	TV Action & Adventure, TV Dramas, TV Sci-Fi & ...	Dark forces conspire against orphan mapmaker A...
7087	Into the Forest	Dramas, International Movies, Sci-Fi & Fantasy	In the near future, two frightened sisters fig...
1779	Marvel's Agents of S.H.I.E.L.D.	TV Action & Adventure, TV Sci-Fi & Fantasy	Agent Phil Coulson, seen in action in Marvel's...

Netflix's recommendation system is a crucial component of its success as a leading streaming service. The system uses a combination of user data, content metadata, and machine learning algorithms to suggest personalized content to users, which has led to increased engagement and retention rates.

The recommendation system employs a variety of techniques, including collaborative filtering, content-based filtering, and deep learning, to generate personalized recommendations for each user. It also uses a multi-armed bandit algorithm to balance exploration and exploitation and optimize the recommendations for user satisfaction.

In conclusion, Netflix's recommendation system plays a vital role in its success and has set the standard for personalized content recommendations in the streaming industry. Its ability to provide tailored content to users has contributed to Netflix's growth and has allowed it to stay ahead of its competitors.

☞ **Role of Each Team Member**

Raymond Sutanto	Project Leader: helped the team for brainstorming, project planning, and data analysis
Cheng Lin Tsai	Helped the team for data exploration
Yi Jen Chen	Helped the team for deploying the result after techniques successfully implemented
Stanley Lin	Helped the team for applying techniques for data analysis